程序设计期末复习

黎金宁 Jinning Li

2016.12

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

第二章

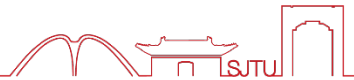
- 宏定义： #define 标识符 替换文本。也可以用来定义常量，C++里定义常量一般使用例如 const int a = 0;
- 变量(对象)包括名称、值、类型。
- 变量名是标识符，标识符以字母或下划线开头，不能是系统保留字，区分大小写。
- 整型：用补码表示，正数的补码是它的二进制表示。负数的补码是将绝对值二进制按位取反后加1，最高位为符号位。
- 10 = 0000000000001010; -10 = 1111111111110110
- 实型：尾数+指数， 1e5, e, 1e3.3, 1.23F
- 枚举类型 enum 枚举类型名 {元素表}
- enum weekday {Sunday=1, Monday, Tuesday, Wednesday=5, Thursday}

三、四章

- sizeof (类型名、数组名) 数组名并不只是首地址，可以通过数组名找到内存大小
- double d; int i; d = i = 1.5
- y=++x; y=x++;
- if, switch, while, break
- 数组：查找：顺序、二分。二分要求数据有序。

```
While （lh <= rh）{
        mid = (lh + rh) / 2;
        If(**) rh = mid -1; else lh = mid + 1;
}
```

第五章



- 排序：
- 直接选择排序
- 冒泡排序
- 快排（分治法）
- 字符串 char ch[**]; cin.getline(字符数组，数组长度，结束标记)
- 结束时遇到结束标记或数组长度-1，因为有'\0'
- cstring strcpy(t,s)复制，strcat(t, s)拼接，strcmp(s1, s2)大于返回正数

第六章

- 函数调用: 系统为函数分配空间，称为帧。局部变量在定义该变量的函数中才有意义。函数结束时空间回收。
- 局部变量， 全局变量， 引用全局的变量 ::p
- 静态全局变量: 函数外， 多个源文件共享, static 限制当前源文件.
- 静态局部变量: 加static存放在全局变量区， 不会消亡
- 二维数组作为参数： 第一维可以省略，第二维必须指定。
- 内联函数：inline 把内联函数的代码复制到调用处，减少函数调用。
- 函数模板：
- `template<class T>`
- `T max(T a,T b){…}`

第六章

- 递归：
- If(终止条件) return (不需要递归的解决方案)
- Else return (包括递归的解决方案)
- 回溯法（八皇后，分书问题)
- 分治法：快排
- 动态规划(硬币找零问题)

第七章

- 指针
- 类型名* 变量名 int* p；
- 引用指针指向的地址：*p
- 指向常量的指针 const int* p = &x; 指针本身可以修改
- 常指针 int* const p = &x；指向的地址的值可以修改
- P = & intarray[1]; 指针运算能自动考虑基本类型的大小
- 动态变量：存在堆内， int *p = new int[10];
- 回收 delete p; delete[] p;
- 内存泄露：new的东西都要删掉

第七章

- 字符串赋给指针：首地址
- void swap(int a, int b) 值传递
- Void swap(int* a, int * b) 调用 swap(&a, &b), 会改变真实值
- 数组传递： 地址传递（数组名代表首地址）
- 引用：给变量取一个别名， 是隐式指针。
- 返回引用：能够通过返回值修改， 如不希望修改用const， 不应返回局部变量。
- 指针数组
- argc, argv; 参数个数， 指向字符的指针数组
- 指向函数的指针： 返回类型(*指针变量)(形式参数表);

第八章

- 结构体
- Struct 类型名{
- 字段声明
- }
- 用'.'逐级访问
- 同一结构体可以互相赋值
- 链表：单链表、双链表、循环链表
- 单链表的插入，删除，头结点：不存放数据，使得不必特判

第十章

- 头文件.h 实现文件.cpp 在实现文件cpp中include其对应的头文件
- 类：数据成员，成员函数

Class 类名{

Private:

私有数据成员;

Public:

公有数据成员;

}

This指针：指向当前对象，　可以使用this-> ; (*this).

第十章

- 构造函数
- 默认构造函数：Complex(){}，或有了指定默认值可以不写默认构造函数。
- 初始化列表：在执行函数体前初始化，提高效率。
- Complex(int a, int b):c=a, d=b{
- .....
- }
- 复制构造函数：类名 (const 类名 &)：对象定义，函数调用，函数返回
- 析构函数：回收对象前自动调用，如delete

第十章

- 常量数据成员：只能在初始化列表中完成
- 静态数据成员：所有对象共用
- 静态成员函数：只能访问静态成员和函数
- 友元：允许某类或函数访问私有成员:
- friend class B;
- friend int B::func(double a);
- 一般写在类开始的位置

第十二章

- 组合
- 继承
- Class 派生类 : [继承方式] 基类名{
- 新增成员声明
- }
- protect 可以被成员函数或友元访问
- Public 继承不可以访问基类的private
- Private 基类成员被继承为Private成员
- 构造与析构：构造从基类开始，析构从派生类开始

第十二章

- 派生类隐式转换为基类对象：
- 赋值
- 用基类指针指向派生类对象
- 基类引用派生类对象 Base &br = d;
- 虚函数：作用是优先执行派生类中的重定义函数
- 虚析构函数：析构函数都被执行, delete基类指针时防止泄漏
- 纯虚函数：没有函数体，定义派生类需要有的行为。

第十三章

- 类模板
- Template<class T> //可以有默认值
- Class 类名{
- }
- 在类外定义需要
- Template<class T>
- Array<T>::****
- 实例化时 类模板名<…> 对象表 例如 Queue Stack
- 普通友元：允许访问所有实例
- 特定友元： friend class B<type>; friend void f(const type &)
- 类模板作为基类派生

Thanks!